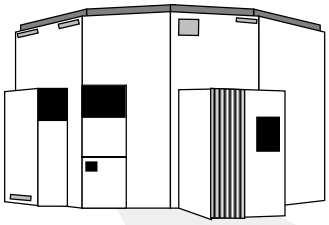
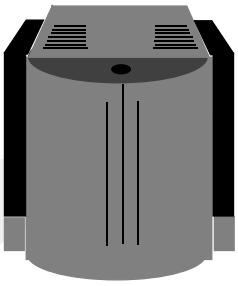
Naval Research Laboratory

Washington, DC 20375-5320





Introduction to AFS

Course Notes

Instructor: Donald Browning
Center for Computational Science
(202)767-3884
donald.browning@nrl.navy.mil

Introduction to AFS

1. Introduction

1.1. What is the Andrew File System?

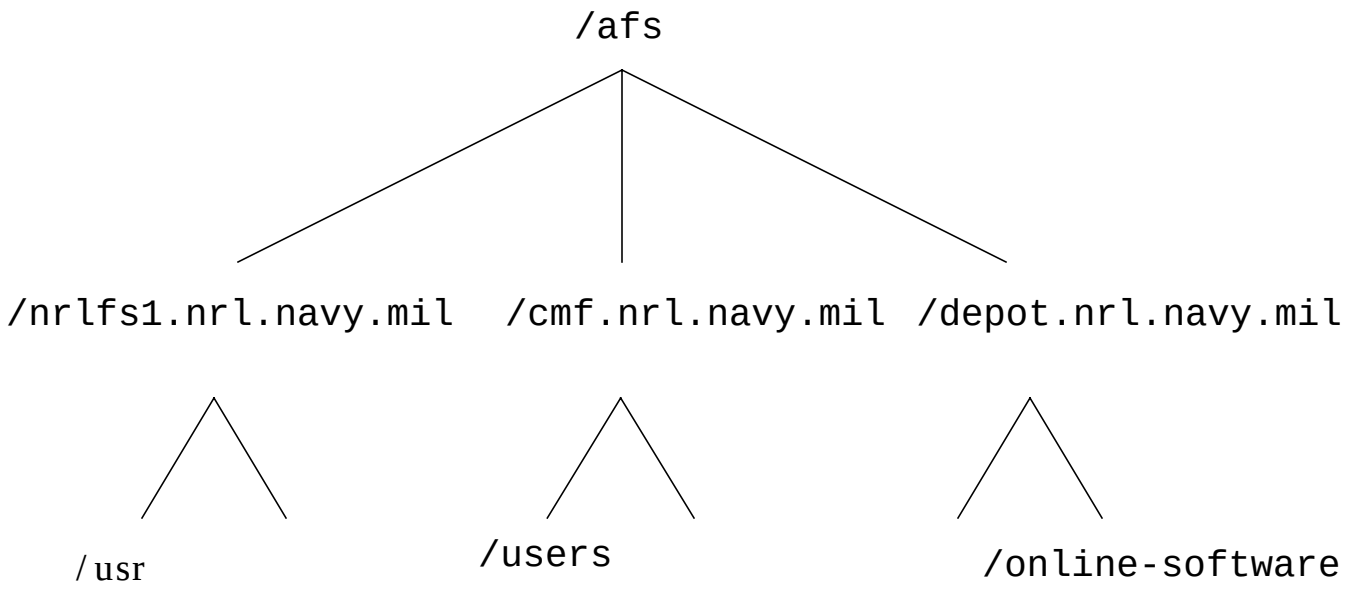
- a distributed file system developed by Carnegie-Mellon University, marketed by Transarc Corp.
- provides transparent file access between systems running AFS
- similar to Sun Microsystems' Network File System (NFS) allows files to be distributed over many systems within an AFS cell
- software available for most UNIX platforms

1.2. Benefits of AFS

- frequently accessed files (e.g. the CCS online software repository) can be stored centrally at a site and accessed from any AFS client
- caching facility improves speed and reduces network traffic
- common name space — a file stored on an AFS server can be accessed using the same filename from any AFS client
- scalability — client / server ratios from 50:1 to 200:1
- improved security — authentication and Access Control Lists (ACLs)
- replicated volumes (more crash resistant)

1.3. AFS File Space Design

- hierarchical file structure like the UNIX file system
- AFS root is generally named `/afs`



- next level is called a cell
  - administrative domain — a defined set of AFS servers within a company, university, laboratory, etc.
  - local cell — the default cell associated with your workstation
  - foreign cell — other cells in the AFS file space
- subsequent levels are UNIX files
- some facilities use AFS for users' login directory

1.4. Client/Server Model

- AFS must run on both client and server systems
- server stores data and transfers it to the client
- client system is the local processor which runs the cache manager, accesses AFS files, and initiates the authentication process

1.4.1. What is the cache manager?

- a process on the client machine which stores ("caches") a copy of the requested file on the client's local disk
- changes are stored when the file closes
- AFS advises cache managers to use most recent versions of a file
- callback is the "promise" that you have the most recent version

1.4.2. What is authentication?

- AFS security process which identifies and authorizes users by granting tokens
- token — a packet of information that is scrambled by the AFS authentication program using your AFS password as a key
- if the Cache Manager can unscramble the token and its information, you are authenticated
- authentication is not always necessary to access AFS space; for example many systems provide files that are readable by any user logged into the AFS client (e.g. CCS online software)

1.5. Online Software

- the CCS provides an AFS server for Online Software
- to participate, your UNIX workstation needs the AFS client
- Computational Support Services (Code 5595) can install the AFS client or, optionally, provide you with the instructions so that you can install it yourself
- the path for the Online Software Project is:

```
/afs/depot.nrl.navy.mil/@sys/bin
```

- when the string '`@sys`' appears in a file name to be used in AFS, it is automatically replaced by the type of the machine on which the file name is being expanded
- this feature is used in pathnames and symbolic links allowing them to be machine independent

2. Authentication . . . Logging On to AFS

AFS provides the tools you'll need to become authenticated (get a token), to change your AFS password, to verify your tokens and to destroy your tokens when your AFS session has ended. Please note that the AFS authentication process is somewhat system-dependent, some sys admins will combine the AFS login with your local system login.

2.1. One-step login

- Login password and authentication password are the same, for example:

```
login: username
Password: password
AFS 3.3 Login
```

2.2. Two-step login

- Login password and authentication password differ
- Authentication done as a separate step, for example:

```
login: username
Password: local_password
. . . message of the day banner . . .
% klog
password: AFS_password
```

Note: Even when you log in using AFS login, you will still need to use `klog`; tokens expire after 25 hours by default.

2.3. Where to Find AFS Authentication Commands

The AFS working space must be in your path in order to access the authentication commands. This is also system-dependent, several examples are included below:

```
/usr/afsws/bin
/afs/cmf.nrl.navy.mil/@sys/bin
/afs/depot.nrl.navy.mil/@sys/bin
```

2.4. Verifying Authentication

You can verify that you have a valid token with the **tokens** command, e.g:

```
% tokens
Tokens held by the Cache Manager:
User's (AFS ID 1011) tokens for afs@cmf.nrl.navy.mil
[Expires May  4 14:41]
User's (AFS ID 1011) tokens for afs@depot.nrl.navy.mil
[Expires May  4 14:41]
User's (AFS ID 1011) tokens for afs@nrlfs1.nrl.navy.mil
[Expires May 15 18:55]
   --End of list--
```

2.5. Changing Your Password

```
% kpasswd
Changing password for 'username' in cell 'cell-name'.
Old password:  old_password
New password (RETURN to abort)  new_password
Retype new password:  new_password
```

2.6. Authenticating in Another Cell

- the `klog` command uses your local cell by default
- to authenticate in a foreign cell, use the following:

```
% klog [-principal username] [-cell cell]
Password:
```

example:

```
% klog -principal moore -cell cmf.nrl.navy.mil
Password:
```

- note that if your username is identical in the foreign cell, then the `-principal` option is unecessary

- to display your local cell name enter:

```
% fs wscell
This workstation belongs to cell 'vl.nrl.navy.mil'
```

- the client machine must recognize a foreign cell in order to login to that cell; to list all available cells:

```
% fs listcells
Cell vl.nrl.navy.mil on hosts nimitz.nrl.navy.mil
     ohm.nrl.navy.mil.
Cell depot.nrl.navy.mil on hosts
     union-station.nrl.navy.mil.
Cell uni-freiburg.de on hosts tibm1.ruf.uni-freiburg.de
     tibm2.ruf.uni-freiburg.de.
Cell anl.gov on hosts antenor.ctd.anl.gov
.
.
.
```

2.7. Unauthenticating and Logging Out

- the lifetime of token is system-dependent
- tokens last for 336 hours on nrlfs1!
- to destroy tokens for the local or specified cell use:

```
% unlog [-cell cell_name]
```

- note that logging out of the client does not accomplish this

3. Accessing Directories and Files in AFS

- use standard UNIX file names and file manipulation commands to access AFS file space
- **`cd`** to access AFS directory
- typical pathname:

`/afs/`*`cell`*`/usr/`*`your_AFS_username`*

examples:

```
% cd /afs/cmf.nrl.navy.mil/users/osburn/ccsnis
% cd /afs/depot/online-software/gamma/@sys/local/bin
```

- **`cp`** to copy files
- **`rm`**, **`rmdir`** to remove files or directories
- **`ls`** to list contents of a directory
- abbreviated cell names are system dependent

3.1. Anonymous AFS users

- not authenticated (no tokens)
- limited (restricted) file and directory access
- assigned to the group **system:anyuser**

3.2. Foreign Cells

- to access files without authenticating (anonymous):

```
% cd /afs/cell
```

- to check which cell a file / directory is located:

```
% fs whichcell
File . lives in cell 'vl.nrl.navy.mil'
```

3.3. AFS/NFS Translator Access

AFS client software is not available for a few UNIX platforms, i.e. the CCF Cray Y-MP EL.

- CCF Cray is NFS mounted to the FS / A
- AFS files can, however, still be accessed via the AFS / NFS translator (nrlafsnfs.nrl.navy.mil) from systems which support NFS

From your local NFS client you would gain access to the AFS file space as follows:

```
% telnet nrlafsnfs

login: username
password: password

$ klog
password: AFS_password

$ knfs -host hostname -id your_uid
.
. (access AFS file space)
.
% knfs -host hostname -id  your_id -unlog

% unlog
$ exit
$ logout
```

`Hostname` is the local system hostname and `your_uid` is your user id on the local system. You will need to exit from the translator shell which is invoked for you.

4. AFS Commands

4.1. Command Structure

AFS commands are grouped into three categories:

- file server commands (fs)
  - lists AFS server information
  - set and list ACLs (access control list)
- protection commands (pts)
  - create and manage (ACL) groups
- authentication commands
  - `klog, unlog`
  - `kpasswd`
  - `tokens`

Table 4.1 AFS Commands

	Command	Alias/Short Form
File Server Commands	fs apropos fs checkservers fs cleanacl fs examine fs help fs listacl fs listcells fs listquota fs quota fs setacl fs whereis fs wscell fs whichcell	fs ap fs checks fs cl fs exa fs h fs la fs listc fs lq fs q fs sa fs whe fs ws fs whi
Protection Commands	pts adduser pts apropos pts chname pts chown pts creategroup pts delete pts examine pts help pts listowned pts membership pts removeuser pts setfields	pts ad pts ap pts chn pts cho pts cg pts del pts e pts h pts listo pts m pts r pts setf
Authentication Commands	/bin/passwd klog kpasswd tokens unlog	

4.2. Command Syntax

```
% command <op_code> -switch <instance> + [-flag]
```

- command ..... command category (i.e. fs or pts)
- op_code ..... command function
- switch ..... option
- instance ..... option argument (may be multiples)
- flag ..... type of output produced
  or how commands execute

example:

```
% fs setacl -dir /afs/cmf/users/browning \
-acl stu09 all -neg
```

- `fs` ..... commande
- `setacl` ..... op code
- `-dir` ..... switch
- *`/afs/cmf/users/browning`*.... instance
- `-acl` ..... switch
- *`stu09`* ..... instance
- `all` ..... switch
- `-neg` ..... flag

4.3. Omitting Switches

- command has only one argument
- all arguments (switch and instance pair) in prescribed order
- switches do not use multiple instances

example:

```
% fs setacl /afs/nrlfs1/usr stu09 all -neg
```

4.4. Short Forms and Aliases

You can type AFS commands in one of 3 ways:

- full command
- short form (any unambiguous abbreviation)
- alias

  - sometimes different from abbreviation
  - see table 4.1

4.5. Online Help

- help and apropos operation codes
- help flag

examples:

```
% fs help listquota
% fs listquota -help
% fs apropos -topic quota
```

5. Listing Information About Your AFS Space

- disks are divided into partitions
- AFS divides partitions into volumes
- access to a volume is through a mount point
- you can move from one volume to another or to a new file server machine transparently
- information you may want to know about your AFS space:
  - size limit (quota) on the volume
  - file server where a file or directory is stored
  - status of the file server machine

5.1. Volume Quota

- cannot store more data than volume quota allows
- if partition is full you may not be able to save data
- three commands to check quota:

`% fs quota` . . . . . lists percentage of volume quota used

`% fs listquota` . . . . lists percentage used of both volume and partition

`% fs examine` . . . . . lists partition's max size, current size, and messages associated with volume

5.2. Location of Files and Directories

- in general, physical location of files and directories is not a concern
- cache manager locates files when you give the pathname
- to determine on which host a file is stored, use:

```
% fs whereis [-path <dir/file path>+]
```

dir / file path	specify the UNIX pathname of each directory or file. Accepts UNIX pathname abbreviations. Can specify multiple(+) directories or files. Default: current directory

example:

```
% fs whereis cm5.manuals
File cm5.manuals is on host picard.cmf.nrl.navy.mil
```

5.3. Status of File Server Machines

Checks if the file server machines in your cell or other cells are operative:

```
% fs checkservers [<cell name>] [-all]
```

examples:

```
% fs checkservers
% fs checkservers -all
% fs checkservers psc.edu
```

6. File Sharing in AFS

- share remote files as easily as local files
- capability to see and share all files under the `/afs` subtree
- file sharing not restricted by geographical distances or operating systems
- cache manager can handle replicas of a file transparently
- How is file sharing done?

  Access Control Lists (ACL)
  AFS Access Rights
  Protection Groups

6.1. Access Control Lists

- allows you to grant or deny access to a directory and its files
- defines 3 special system groups
- AFS interprets only the UNIX "user" mode bits for file protection

6.1.1. Access Rights

- seven standard access rights
- each right has a single-character abbreviation
- access rights divided into 2 groups

  directory rights — apply to the directory
  file rights — apply to the files in a directory

6.1.2. Directory rights

1. LOOKUP (l) . . . . . read contents of a directory and examine the ACL. Must have this right before you can access files or directories in any other way
2. INSERT (i) . . . . . add new files or create new subdirectories
3. DELETE (d) . . . . . remove files / subdirectories
4. ADMINISTER (a) . . . Change ACL for the directory. You always have this right on the home directory

6.1.3. File Rights

5. READ (r) . . . . . read contents of the directory and file data
6. WRITE (w) . . . . . modify file and change UNIX mode bits (chmod)
7. LOCK (k). . . . . run progams that need to "flock" files

6.1.4. Shorthand

1. WRITE . . . . . all rights except ADMINISTER (rlidwk)
2. READ. . . . . READ and LOOKUP rights (rl)
3. ALL . . . . . all seven rights (rlidwka)
4. NONE . . . . . no rights; removes user's entry from the ACL

6.1.5. Normal and Negative Rights

- normal rights .... **grant** access to a user or group
- negative rights ... explicitly **deny** access to a user or group

6.1.6. System-defined groups

- `system:anyuser` ... everyone who can access your cell
- `system:authuser` .. everyone "authenticated" in your cell
- `system:administrators` ..... a select few designated by the system admininstrator

6.2. ACL Commands

6.2.1. Listing an ACL

```
% fs listacl [-path <dir/file path>+]
```

- dir/file path ..... specify directory whose ACL is to be listed. Can specify multiple(+) directories or files separated by spaces. Default: current directory
- example:

```
% fs listacl /afs/cmf/users/browning
Access list for /afs/cmf/users/browning is
Normal rights:
  system:administrators rl
  system:anyuser rl
  browning rlidwka
Negative rights:
  stu09 rlidwka
```

6.2.2. Granting Access with an ACL

```
% fs setacl -dir <directory>+  \
     -acl <access list entries>+
```

- directory ..... Directory whose ACL is to be modified. Separate multiple(+) directories with spaces.

- access list entries . . . Pairs of users / groups and rights (in that order, separated by a space)

example:

```
% mkdir test
% fs listacl test
Access list for test is
Normal rights:
  system:administrators rl
  system:anyuser rl
  browning rlidwka
% fs setacl test osburn rli system:anyuser rli
% fs listacl test
Access list for test is
Normal rights:
  system:administrators rl
  system:anyuser rli
  osburn rli
  browning rlidwka
```

6.2.3. Changing an ACL

- use the `fs setacl` command
- command uses `-dir` and `-acl` switches to separate directory names from the list of access list entries
- command flags give or take away normal or negative rights

6.3. Using Protection Groups

- A list of individual users ("group") you can place on an ACL
- Creator is automatically the owner
- Owner is only one allowed to add members, remove members, rename the group, change the group owner or delete the group

6.3.1. Uses of Groups

- private use — created for your own use
- shared use — others can add the group to their ACLs
- group names are in the format
  `owner-name:group-name`
- system administrator sets the group quota

6.3.2. Listing Information about Groups

- to determine who belongs to a group:

```
% pts membership -nameorid \
      <user or group name or id>+
```

-nameorid . . . . . specify the complete name or AFS UID of each group about which information is to be displayed. Separate multiple names of AFS UIDs with spaces

examples:

```
% pts membership moore:this_class
% pts m browning -cell cmf
Groups browning (id: 1011) is a member of:
osburn:sysadmins
```

- to determine who owns or created a group:

```
% pts examine -nameorid \
      <user or group name or id>+
```

example:

```
% pts examine moore:this_class
```

- to determine the groups that a group owns:

```
% pts listowned -nameorid \
      <user or group name or id>+
```

example:

```
% pts listowned moore:this_class
```

6.3.3. Creating Groups and Adding Members

- to create a group in your cell:

```
% pts creategroup -name <group name>+ \
      [-owner <owner of group>]
```

`-name` ..... must be of the form owner-name:group-name

`-owner` .... use if someone else should own the group

example:

```
% pts creategroup stu00:mygroup
```

- groups cannot be members of groups
- to add members to a group you own:

```
% pts adduser -user <user name>+ \
      -group <group name>+
```

`-user`..... user name of the person to be added

`-group` .... complete name of group to which -user is added

example:

```
% pts adduser -user stu01 stu02 \
      stu03 -group stu00:mygroup
```

7. Additional AFS Resources

This course is intended as an introduction to and brief overview of AFS. There is a wealth of additional information available on the Internet. The AFS **faq** (frequently asked questions) is the best place to start. This document is about 30 pages in length and is available via several methods:

- via FTP: ftp://ftp.transarc.com/pub/afs-contrib/doc/faq/afs.faq
- via WWW:http://www.transarc.com/Product/AFS/FAQ/faq.html
- via email:
  mail -s afs.faq auto-send@mailserver.aixssc.uk.ibm.com < /dev/null

The Usenet newsgroup:

- **alt.filesystems.afs**

discusses AFS. There is also an AFS user group, check out **pub/afsug** via anonymous ftp at **grand.central.org** (see README for details).

World Wide Web users can find links to the faq, a beginner's guide, AFS Users Group newsletters, etc. by accessing:

- http://www.cs.cmu.edu/afs/andrew.cmu.edu/usr/db74/www/afs.html

Of course, there is also an AFS mailing list available at **info-afs@transarc.com.** See the faq for instructions on how to subscribe to this list and for the location of its archives.